Upgrading Platform **LSF®** on UNIX and Linux

Version 6.0
November 28 2003
Platform Computing

Comments to: doc@platform.com

Contents

- "Which Upgrade Steps to Use"
- "Upgrading an LSF Cluster Installed with lsfinstall"
- "Migrating an Existing Cluster to the lsfinstall Directory Structure"
- "Compatibility Notes"
- "Getting Technical Support"
- "Copyright"

Platform™

Which Upgrade Steps to Use

Use this document to upgrade your Platform **LSF**® installation ("LSF") to Version 6.0.

`lsfsetup` is no longer supported for installing or upgrading LSF. You must use `lsfinstall` with one of the following procedures to upgrade your cluster:

- "Upgrading an LSF Cluster Installed with lsfinstall"
  Use this procedure if you used `lsfinstall` to install your cluster.
- "Migrating an Existing Cluster to the lsfinstall Directory Structure"
  Use this procedure to migrate an LSF cluster installed or upgraded with `lsfsetup` to the LSF directory structure supported by `lsfinstall` in LSF Version 4.2 and later.

Upgrading an LSF Cluster Installed with lsfinstall

Use this procedure if you used `lsfinstall` to install your cluster.

If your cluster was previously installed or upgraded with `lsfsetup`, DO NOT use these steps. Use the steps in "Migrating an Existing Cluster to the lsfinstall Directory Structure".

Contents

- "Before you upgrade"
- "Download LSF distribution tar files"
- "Use lsfinstall to upgrade LSF"
- "Use hostsetup to set up LSF hosts"

Before you upgrade

1. You should inactivate all queues to make sure that no new jobs will be dispatched during the upgrade. After upgrading, remember to activate the queues again so pending jobs can be dispatched.
   - To inactivate all LSF queues, use the following command:
     ```
     % badmin qinact all
     ```
   - To reactivate all LSF queues after upgrading, use the following command:
     ```
     % badmin qact all
     ```
2. Before using this procedure, back up your existing LSF_CONFDIR, LSB_CONFDIR, and LSB_SHAREDIR according to the procedures at your site.
3. Get an LSF Version 6.0 license and create a license file (`license.dat`).

Download LSF distribution tar files

1. Log on to the LSF file server host as `root`.
2. FTP to `ftp.platform.com` and get the following files from the `/distrib/6.0/platform_lsf/` directory on `ftp.platform.com`:
   - LSF installation script tar file `lsf6.0_lsfinstall.tar.Z`
   - LSF distribution tar files for all host types you need

     Put the distribution tar files in the same directory as `lsf6.0_lsfinstall.tar.Z`.

   Download and read the LSF Version 6.0 `readme.html` and `release_notes.html` files for detailed steps for downloading LSF distribution tar files.
3. Uncompress and extract `lsf6.0_lsfinstall.tar.Z`:
   ```
   # zcat lsf6.0_lsfinstall.tar.Z | tar xvf -
   ```

IMPORTANT **DO NOT extract the distribution tar files.**

Use lsfinstall to upgrade LSF

1. Change to `lsf6.0_lsfinstall/`.
2. Read `lsf6.0_lsfinstall/install.config` and decide which installation variables you need to set.
3. Edit `lsf6.0_lsfinstall/install.config` to set the installation variables you need.
4. Follow the instructions in `lsf_unix_install_6.0.pdf` to run:

   ```
   # ./lsfinstall -f install.config
   ```

IMPORTANT You must run `lsfinstall` as `root`.

`lsfinstall` backs up the following configuration files for your current installation in LSF_CONFDIR:

- `cshrc.lsf`
- `lsf.cluster.`*`cluster_name`*
- `lsf.conf`
- `lsf.shared`
- `profile.lsf`

Use hostsetup to set up LSF hosts

1. Follow the steps in `lsf6.0_lsfinstall/lsf_getting_started.html` to set up your LSF hosts (`hostsetup`).
   a. Log on to each LSF server host as `root`. Start with the LSF master host.
   b. Run `hostsetup` on each LSF server host. For example:

      ```
      # cd /usr/share/lsf/6.0/install
      # ./hostsetup --top="/usr/share/lsf/"
      ```

      For complete `hostsetup` usage, enter `hostsetup -h`.
2. Set your LSF environment:
   - For `csh` or `tcsh`:

     ```
     % source LSF_TOP/conf/cshrc.lsf
     ```

   - For `sh`, `ksh`, or `bash`:

     ```
     $ . LSF_TOP/conf/profile.lsf
     ```
3. Follow the steps in `lsf6.0_lsfinstall/lsf_quick_admin.html` to update your license.
4. Use the following commands to shut down the old LSF daemons:

   ```
   % badmin hshutdown all
   % lsadmin resshutdown all
   % lsadmin limshutdown all
   ```
5. Use the following commands to restart LSF using the new 6.0 daemons:

   ```
   % lsadmin limstartup all
   % lsadmin resstartup all
   % badmin hstartup all
   ```
6. Follow the steps in `lsf6.0_lsfinstall/lsf_quick_admin.html` to verify that your upgraded cluster is operating correctly.
7. Use the following command to reactivate all LSF queues after upgrading:

```
% badmin qact all
```

8 Have users run one of the LSF shell environment files to switch their LSF environment to the new cluster.

Follow the steps in `lsf6.0_lsfinstall/lsf_quick_admin.html` for using `LSF_CONFDIR/cshrc.lsf` and `LSF_CONFDIR/profile.lsf` to set up the LSF environment for users.

After the new cluster is up and running, users can start submitting jobs to it.

Migrating an Existing Cluster to the lsfinstall Directory Structure

Use this procedure to migrate an LSF cluster installed or upgraded with `lsfsetup` to the LSF directory structure supported by `lsfinstall` in LSF Version 4.2 and later.

If your cluster was installed with lsfinstall, DO NOT use these steps. Use the steps in "Upgrading an LSF Cluster Installed with lsfinstall" to upgrade your cluster.

Contents

- "Before you upgrade"
- "Download LSF distribution tar files"
- "Use lsfinstall to install an independent LSF 6.0 cluster"
- "Use hostsetup to set up LSF hosts"
- "Migrate the configuration files from existing cluster"
- "Migrate customized commands in LSF_BINDIR from existing cluster"
- "Migrate external executables in LSF_SERVERDIR from existing cluster"
- "Migrate integrations and special setup from existing cluster"
- "Bring the new cluster online"

Before you upgrade

1. You should inactivate all queues to make sure that no new jobs will be dispatched during the upgrade. After upgrading, remember to activate the queues again so pending jobs can be dispatched.
   - To inactivate all LSF queues, use the following command:

     ```
     % badmin qinact all
     ```
   - To reactivate all LSF queues after upgrading, use the following command:

     ```
     % badmin qact all
     ```
2. Before using this procedure, back up your existing LSF_CONFDIR, LSB_CONFDIR, and LSB_SHAREDIR according to the procedures at your site.
3. Get an LSF Version 6.0 license and create a license file (`license.dat`).

Download LSF distribution tar files

1. Log on to the LSF file server host as `root`.
2. FTP to `ftp.platform.com` and get the following files from the `/distrib/6.0/platform_lsf/` directory on `ftp.platform.com`:
   - LSF installation script tar file `lsf6.0_lsfinstall.tar.Z`
   - LSF distribution tar files for all host types you need

     Put the distribution tar files in the same directory as `lsf6.0_lsfinstall.tar.Z`.

Download and read the LSF Version 6.0 readme.html and release_notes.html files for detailed steps for downloading LSF distribution tar files.

3 Uncompress and extract lsf6.0_lsfinstall.tar.Z:

```
# zcat lsf6.0_lsfinstall.tar.Z | tar xvf -
```

DO NOT extract the distribution tar files.

Use lsfinstall to install an independent LSF 6.0 cluster

1 Change to lsf6.0_lsfinstall/.

2 Read lsf6.0_lsfinstall/install.config and decide which installation variables you need to set.

3 Edit lsf6.0_lsfinstall/install.config to set the installation variables you need.

If your cluster uses scripts that depend on having LSF_BINDIR, LSF_SERVERDIR, and LSF_LIBDIR configured in lsf.conf, set a value for UNIFORM_DIRECTORY_PATH to machine-dependent files in lsf6.0_lsfinstall/install.config.

For example, if your current configuration is:

- LSF_BINDIR="/usr/share/lsf/bin"
- LSF_SERVERDIR="/usr/share/lsf/etc"
- LSF_LIBDIR="/usr/share/lsf/lib"

Then set:

```
UNIFORM_DIRECTORY_PATH="/usr/share/lsf"
```

4 Follow the instructions in lsf_unix_install_6.0.pdf to run:

```
# ./lsfinstall -f install.config
```

IMPORTANT You must run lsfinstall as root.

Use hostsetup to set up LSF hosts

1 Follow the steps in lsf6.0_lsfinstall/lsf_getting_started.html to set up your LSF hosts (hostsetup).

a Log on to each LSF server host as root. Start with the LSF master host.

b Run hostsetup on each LSF server host. For example:

```
# cd /usr/share/lsf/6.0/install
# ./hostsetup --top="/usr/share/lsf/"
```

For complete hostsetup usage, enter hostsetup -h.

2 Set your LSF environment:

- For csh or tcsh:

```
% source LSF_TOP/conf/cshrc.lsf
```

- For sh, ksh, or bash:

```
$ . LSF_TOP/conf/profile.lsf
```

3 Follow the steps in lsf6.0_lsfinstall/lsf_quick_admin.html to update your license.

Migrate the configuration files from existing cluster

LSF_CONFDIR

1. Add configuration parameters from existing `lsf.conf` to the new `lsf.conf`.
2. Merge the licensed features in the PRODUCTS line of the existing `lsf.cluster.`*`cluster_name`* into the new `lsf.cluster.`*`cluster_name`*.

   For example, if your existing `lsf.cluster.`*`cluster_name`* file has the the following PRODUCTS line:

```
PRODUCTS=LSF_Base LSF_Batch LSF_Make LSF_MultiCluster
```

   and your new file has the following PRODUCTS line:

```
PRODUCTS=LSF_Base LSF_Manager LSF_Sched_Fairshare LSF_Sched_Preemption
LSF_Sched_Resource_Reservation LSF_MultiCluster
```

   Remove the LSF_Batch feature, and add the `LSF_Make` feature to the PRODUCTS line in the new `lsf.cluster.`*`cluster_name`* file:

```
PRODUCTS=LSF_Base LSF_Manager LSF_Sched_Fairshare LSF_Sched_Preemption
LSF_Sched_Resource_Reservation LSF_Make LSF_MultiCluster
```

3. Copy the following files from the existing LSF_CONFDIR to the new LSF_CONFDIR:
   - `lsf.task`
   - `lsf.shared`
   - `hosts`, if it exists

LSB_CONFDIR

Copy the following files from the existing `LSB_CONFDIR/`*`cluster_name`*`/configdir/` to the new `LSB_CONFDIR/`*`cluster_name`*`/configdir/`:

- `lsb.hosts`
- `lsb.params`
- `lsb.queues`
- `lsb.users`

Migrate customized commands in LSF_BINDIR from existing cluster

Copy any customized LSF command wrappers to the new LSF_BINDIR.

For example:

```
# mv /usr/share/lsf/6.0/sparc-sol7-32/bin/bsub
/usr/share/lsf/6.0/sparc-sol7-32/bin/bsub.real
```

```
# cp /usr/share/lsf/4.1/sparc-sol7-32/bin/bsub
/usr/share/lsf/6.0/sparc-sol7-32/bin/bsub
```

See the *Platform LSF Reference* to verify that the command-line options of your command wrappers are still available.

Migrate external executables in LSF_SERVERDIR from existing cluster

Copy the following files in LSF_SERVERDIR of the existing cluster to the new LSF_SERVERDIR under LSF_TOP:

- `esub`
- `elim`
- `egroup`

Copy any other customized external executables to the new LSF_SERVERDIR.

For example:

```
# cp /usr/share/lsf/4.1/sparc-sol7-32/etc/eexec
/usr/share/lsf/6.0/sparc-sol7-32/etc/eexec
# cp /usr/share/lsf/4.1/sparc-sol7-32/etc/erestart
/usr/share/lsf/6.0/sparc-sol7-32/etc/erestart
```

Migrate integrations and special setup from existing cluster

- If you use any LSF integrations, you should reinstall all integration packages for LSF Version 6.0.
- Do any special setup procedures; for example, TRIX installation, as the final migration step.

Bring the new cluster online

On the existing cluster

1. Use the command

   ```
   % badmin qclose all
   ```

   to close all queues.
2. Notify users to stop submitting jobs to the existing cluster.
3. After all jobs have finished running on the existing cluster, use `lsfshutdown` to shut down the cluster.

On the new cluster

1. Set your LSF environment:
   - For `csh` or `tcsh`:

     ```
     % source LSF_TOP/conf/cshrc.lsf
     ```
   - For `sh`, `ksh`, or `bash`:

     ```
     $ . LSF_TOP/conf/profile.lsf
     ```
2. Use `lsfstartup` to start the new cluster.
3. Use the following command to reactivate all LSF queues after upgrading:

   ```
   % badmin qact all
   ```
4. Have users run one of the LSF shell environment files to switch their LSF environment to the new cluster.

   Follow the steps in `lsf6.0_lsfinstall/lsf_quick_admin.html` for using `LSF_CONFDIR/cshrc.lsf` and `LSF_CONFDIR/profile.lsf` to set up the LSF environment for users.

   After the new cluster is up and running, users can start submitting jobs to it.

Compatibility Notes

API Compatibility between LSF 5.x and Version 6.0

Full backward compatibility: your applications will run under LSF Version 6.0 without changing any code.

The Platform LSF Version 6.0 API is fully compatible with the LSF Version 5.*x* and Version 4.*x* API. An application linked with the LSF Version 5.*x* and Version 4.*x* library will run under LSF Version 6.0 without relinking.

To take full advantage of new Platform LSF Version 6.0 features, you should recompile your existing LSF applications with LSF Version 6.0.

Server host compatibility Platform LSF

You must upgrade the LSF master hosts in your cluster to Version 6.0.

LSF 5.*x* servers are compatible with Version 6.0 master hosts. All LSF 5.x features are supported by 6.0 master hosts except:

To use new features introduced in Platform LSF Version 6.0, you must upgrade all hosts in your cluster to 6.0.

Platform LSF MultiCluster

You must upgrade the LSF master hosts in all clusters to Version 6.0.

New configuration parameters and environment variables

The following new parameters and environment variables have been added for LSF Version 6.0:

lsb.hosts

EXIT_RATE specifies a threshold in minutes for exited jobs

lsb.params

- EADMIN_TRIGGER_DURATION defines how often `LSF_SERVERDIR/eadmin` is invoked once a job exception is detected.
- JOB_EXIT_RATE_DURATION defines how long LSF waits before checking the job exit rate for a host.
- ABS_RUNLIMIT—if set, the run time limit specified by the `-W` option of `bsub`, or the RUNLIMIT queue parameter in `lsb.queues` is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted with a run limit.

lsb.queues

- DISPATCH_ORDER defines an ordered cross-queue fairshare set
- JOB_IDLE specifies a threshold for idle job exception handling
- JOB_OVERRUN specifies a threshold for job overrun exception handling
- JOB_UNDERRUN specifies a threshold for job underrun exception handling
- RES_REQ accepts multiple ptile specifications in the span section for dyamic ptile enforcement
- SLOT_POOL is the name of the pool of job slots the queue belongs to for queue-based fairshare

- SLOT_SHARE specifies the share of job slots for queue-based fairshare, representing the percentage of running jobs (job slots) in use from the queue
- THREADLIMIT limits the number of concurrent threads that can be part of a job. Exceeding the limit causes the job to terminate
- RUNLIMIT—if ABS_RUNLIMIT=Y is defined in `lsb.params`, the run time limit is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted to a queue with a run limit configured.

Environment variables

- LSB_SUB_EXTSCHED_PARAM
  Value of external scheduling options specified by `bsub -extsched`, or queue-level MANDATORY_EXTSCHED or DEFAULT_EXTSCHED
- LSB_SUB_JOB_WARNING_ACTION
  Value of job warning action specified by `bsub -wa`
- LSB_SUB_JOB_WARNING_TIME_PERIOD
  Value of job warning time period specified by `bsub -wt`

New command options and output

The following command options and output have changed for LSF Version 6.0:

bacct

- `-sla` *service_class_name* displays accounting statistics for jobs that ran under the specified service class
- `-x` displays jobs that have triggered a job exception (overrun, underrun, idle)

badmin

- `schddebug` sets message log level for `mbschd` to include additional information in log files
- `schdtime` sets timing level for `mbschd` to include additional timing information in log files
- `-C` *comment* logs the text of *comment* as an administrator comment record to `lsb.events` for the following subcommands:
  - `mbdrestart`
  - `qopen`
  - `qclose`
  - `qact`
  - `qinact`
  - `hopen`
  - `hclose`

bhist

`-l` displays:

- Job group modification
- Configured thread limit

bhosts

- `-x` displays hosts whose job exit rate has exceeded the threshold configured by EXIT_RATE in `lsb.hosts` for longer than JOB_EXIT_RATE_DURATION configured in `lsb.params`, and are still high

- `-l` displays the comment text if the LSF administrator specified an administrator comment with the `-C` option of the `badmin` host control commands `hclose` or hopen

bjobs

- `-g` *job_group_name* displays information about jobs attached to the specified job group
- `-l` displays the thread limit for the job
- `-sla` *service_class_name* displays jobs belonging to the specified service class
- `-x` displays unfinished jobs that have triggered a job exception (overrun, underrun, idle)

bkill

- `-g` *job_group_name* operates only on jobs in the specified job group
- `-sla` *service_class_name* operates on jobs belonging to the specified service class.

bmod

- `-g` *job_group_name* | `-gn`
- `-sla` *service_class_name* | `-slan`

bqueues

`-l` displays:

- Configured job exception thresholds and number of jobs in each exception state for the queue
- The job slot share (SLOT_SHARE) and the name of the share pool (SLOT_POOL) that the queue belongs to for queue-based fairshare
- DISPATCH_ORDER in a master queue for cross-queue fairshare
- The comment text if the LSF administrator specified an administrator comment with the `-C` option of the queue control commands `qclose`, `qopen`, `qact`, and `qinact`, `qhist`

bresume

`-g` *job_group_name* resumes only jobs in the specified job group

brsvadd

`-R` selects hosts for the reservation according to the specified resource requirements

bstop

- `-g` *job_group_name* suspends only jobs in the specified job group
- `-sla` *service_class_name* suspends jobs belonging to the specified service class

bsub

- `-g` *job_group_name* submits jobs in the specified job group
- `-R` accepts multiple ptile specifications in the span section for dyamic ptile enforcement
- `-sla` *service_class_name* specifies the service class where the job is to run
- `-T` *thread_limit* sets the limit of the number of concurrent threads to thread_limit for the whole job.
- `-W`—if ABS_RUNLIMIT=Y is defined in `lsb.params`, the run time limit is not normalized by the host CPU factor. Absolute wall-clock run time is used for all jobs submitted with a run limit.

New files added to installation

The following new files have been added to the Platform LSF Version 6.0 installation:

- `LSB_CONFDIR/`*`cluster_name`*`/configdir/lsb.serviceclasses`
- `LSF_BINDIR/bgadd`
- `LSF_BINDIR/bgdel`
- `LSF_BINDIR/bjgroup`
- `LSF_BINDIR/blimits`
- `LSF_BINDIR/bsla`
- `LSF_SERVERDIR/eadmin`
- `LSF_LIBDIR/schmod_jobweight.so`

Symbolic links to LSF files

If your installation uses symbolic links to other files in these directories, you must manually create links to these new files.

New accounting and job event fields

The following fields have been added to `lsb.acct` and `lsb.events`:

lsb.acct

- JOB_FINISH:

  sla (%s) is the SLA service class name under which the job runs.

lsb.events

- JOB_NEW:
  - sla (%s) is the SLA service class name under which the job runs
  - SLArunLimit (%d) is the absolute run time limit of the job for SLA service classes
  - jobGroup (%s) is the job group under which the job runs
- JOB_MODIFY2:
  - sla (%s) is the SLA service class name that the job is to be attached to
  - jobGroup (%s) is the job group under which the job runs
- JOB_EXECUTE:

  SLAscaledRunLimit (%d) is the run time limit for the job scaled by the execution host
- QUEUE_CTRL:

  ctrlComments (%s) is the administrator comment text from the `-C` option of `badmin` queue control commands `qclose`, `qopen`, `qact`, and `qinact`
- HOST_CTRL:

  ctrlComments (%s) is the administrator comment text from the `-C` option of `badmin` host control commands `hclose` and `hopen`
- MBD_DIE:

  ctrlComments (%s) is the administrator comment text from the `-C` option of `badmin mbdrestart`

Version 4.x license features

A permanent LSF license allows only one FEATURE line for each LSF product or feature. If your license file is used by multiple LSF clusters, and you want to upgrade just one cluster, you have to upgrade the licenses all at once.

For example, the 5.*x* and 6.0 FEATURE line for `lsf_base` replaces the 4.*x* FEATURE line for `lsf_base`. However, version 5.*x* and 6.0 licenses are not fully compatible with LSF version 4.*x* licenses. The 4.*x* `lsf_batch` feature is not included in 5.*x* and 6.0 licenses. To use one license file to run 4.*x,* 5.*x*, and 6.0 clusters, you must add the 4.*x* `lsf_batch` feature to your 5.x and 6.0 licenses.

To make your license work for all versions of LSF, you must manually edit the 5.*x* and 6.0 license files and append the 4.*x* FEATURE line for `lsf_batch`, and also any 4.*x* INCREMENT lines for `lsf_batch`.

After you upgrade

After all your clusters have been upgraded from LSF Version 4.x, you can delete the lsf_batch lines from your license file. Always reconfigure the cluster after upgrading your license file.

Getting Technical Support

Contacting Platform

Contact Platform Computing or your LSF vendor for technical support. Use one of the following to contact Platform technical support:

Email support@platform.com

World Wide Web www.platform.com

Phone
- North America: +1 905 948 4297
- Europe: +44 1256 370 530
- Asia: +86 10 6238 1125

Toll-free phone 1-877-444-4LSF (+1 877 444 4573)

Mail
Platform Support
Platform Computing
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

When contacting Platform, please include the full name of your company.

We'd like to hear from you

If you find an error in any Platform documentation, or you have a suggestion for improving it, please let us know:

Email doc@platform.com

Mail
Information Development
Platform Computing
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

Be sure to tell us:
- The title of the manual you are commenting on
- The version of the product you are using
- The format of the manual (HTML or PDF)

Copyright

Document redistribution policy

Internal redistribution

Trademarks